



**HAL**  
open science

## A robust set approach for mobile robot localization in ambient environment

Etienne Colle, Simon Galerne

► **To cite this version:**

Etienne Colle, Simon Galerne. A robust set approach for mobile robot localization in ambient environment. *Autonomous Robots*, 2019, 43 (3), pp.557–573. 10.1007/s10514-018-9727-4 . hal-01758277

**HAL Id: hal-01758277**

**<https://hal.science/hal-01758277>**

Submitted on 4 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A robust set approach for mobile robot localization in ambient environment

Etienne Colle · Simon Galerne

Etienne.Colle@Ibisc.univ-evry.fr  
Simon.Galerie@Ibisc.univ-evry.fr  
Laboratory IBISC), University of Evry, 91000 Evry, France

**Abstract** Mobile robot localization consists in estimating the robot coordinates using real-time measurements. In ambient environment context, data can come both from the robot on-board sensors and from environment objects, mobile or not, able to sense the robot. The paper considers localization problem as a nonlinear bounded-error estimation of the state vector. The components of the state vector are the robot coordinates as well as the 2D position and orientation. The approach based on interval analysis can satisfy the needs of ambient environment by easily taking account a heterogeneous set and a variable number of measurements. Bounded-error state estimation can be an alternative to particle filtering which is sensitive to non-consistent measures, large measure errors, and drift of evolution model. The paper addresses the theoretical formulation of the set-membership approach and the application to the estimation of the robot localization. Additional treatments are added to the estimator in order to meet more realistic conditions. Treatments aim at reducing the effects of disruptive events: outliers, model inaccuracies or model drift and robot kidnapping. Simulation results show the contribution of each step of the estimator.

**Keywords** Robot mobile localization · Interval analysis · Bounded-error · Outlier, Model inaccuracy and drift · Multi-hypothesis tracking

## 1 Introduction

Autonomous navigation of mobile robots needs to continuously estimate the pose  $(x_R, y_R, \theta_R)$  which are the position and the orientation of the vehicle, in a given frame of reference. This localization problem has been deeply studied in the last decades. Several strategies have been proposed providing the estimate of robot pose once an environment map is available.

Two different approaches can be distinguished local localization and global localization. Local localization or position tracking provides a new pose estimate, given a previous position and proprioceptive information. However an accurate estimate of the initial pose is required. Global localization is designed to estimate the robot pose without any a priori given exteroceptive information. Global localization is used for initializing the pose estimate of the robot or for solving the lost robot problem. Proprioceptive and exteroceptive information is usually combined, which leads to classify localization strategies into three categories (Filliat and Meyer 2003):

- Strategies which directly deduce the pose from exteroceptive data only. They are adapted to global localization in environments with no perceptual aliasing. Perceptual aliasing means that different places can be recognized while perception being the same.
- Strategies which track a single hypothesis about robot pose using both exteroceptive and proprioceptive information. They allow position tracking in environments with perceptual aliasing. However if the previous estimation becomes wrong due to model derive or robot drop off, pose updating will be incorrect and the robot gets lost.
- Strategies which track multiple hypotheses about robot pose using both exteroceptive and proprioceptive information. The set of hypotheses are updated in parallel. They make global positioning possible in environments with perceptual aliasing. In addition they are less sensitive to the drift of models and to outliers.

The last two localization strategies are generally considered as a state estimation problem (Paull et al. 2014). Let a robot pose at time  $t$  given by  $\mathbf{x}(t)$ , the goal of recursive state estimation is to estimate the belief distribution  $Bel$  of the state  $\mathbf{x}(t)$ . The propagation of the state is given by some general non-linear state equation. Typically the state at time  $t$  is recursively estimated through an approximation of the Bayes filter which consists of two components: prediction and innovation. This formulation assumes the Markov hypothesis which states that only the most recent estimates, control and measurements need to be considered to generate the estimate of the next state. In fact, Bayesian filtering includes a large number of approaches that differ by the way the probability distribution of  $Bel(\mathbf{x}(t))$  is represented. In Kalman filter,  $Bel(\mathbf{x}(t))$  is represented by a Gaussian probability distribution. In Multiple-Kalman filter used for managing multiple hypotheses,  $Bel(\mathbf{x}(t))$  is represented by a Gaussian mixture distribution, and in particle filter by a set of samples. Extended Kalman Filter (EKF) has been early investigated for solving localization problem (Cox 1991; Dissanayake et al. 2001; Leonard et al. 1992; Castellanos et al. 2004). However, as in many cases real-world disturbances do not satisfy statistical assumptions, the EKF is not guaranteed to converge. In addition, the tuning of parameters can be difficult and EKF needs an accurate initialization. Tracking multiple hypotheses by multiple Kalman filters allows significant improvement but at the expense of increased complexity (Piasecki 1995). EKF drawbacks have led to the development of other filters. The particle filtering (PF) is one of the most effective localization algorithms (Thrun et al. 2000; Gustafson 2010). PF is based on the sequential Monte Carlo method, designed to solve the optimal filtering problem numerically. The posterior distribution of the state is approximated by a large set of Dirac-delta masses (particles) that evolve randomly in time according to the dynamics of the model and the observations. PF advantage is that non-Gaussian distributions and non-linear models can be incorporated. PF is known for its robustness and the ease of implementation if the state dimension is low, but might also be difficult to tune. In fact the efficiency of the filter depends mostly on the number of particles and on the way to re-allocate the weights of particles. The number of particles results from a compromise between robustness, accuracy and computing time. The number of particles is chosen to be large enough to achieve good behavior in the start-up phase and to increase robustness. However, in the normal operational mode the number of particles can be decreased substantially (typically by a factor of ten).

According to Gustafson (2010) who has surveyed PF theory and practice for positioning applications, PF is not practically useful when extending the models to more realistic cases. Three main cases are mentioned:

1. High-dimensional state-space models, typically involving motion in more than three-dimensions space (six-dimensional pose)
2. More dynamic states (including accelerations, unmeasured velocities)
3. Sensor bias and model drift

The Marginalized Particle Filter (MPF) brings an answer to the case of high-dimensional state-space models as long as the nonlinearities only affect a small subset of the state. This way, the particle filter is used to solve the high nonlinearities of the state. (E)KF are used for the (almost) linear Gaussian states. The fastSLAM algorithm is in fact a version of the MPF (Dissanayake et al. 2011). PF are also sensitive to non-consistent measures, large measure errors (Lambert et al. 2009) and model drift. Experiments point out that PF can locally converge towards a wrong solution due to erroneous measurements. Similar experiments with an EKF show the same phenomenon. EKF strongly underestimates its covariance matrix in presence of repeated biased measurements Castellanos et al. (2004). Tracking multiple hypotheses by multiple Kalman filters allows significant improvement but at the expense of increased complexity (Piasecki 1995).

An alternative to PF could be the bounded-error state estimation. Experiments have demonstrated that the approach is operational on vehicle navigating in an indoor environment as well as outdoor environment (Seigniez et al. 2009). In this approach, known as set-membership or set-theoretic estimation, all model and measurement errors are assumed to be bounded, with known bounds. The set-membership approach represents the solution of a problem by a closed set in which the real solution is guaranteed to be, in the sense that no solution consistent with the data and the hypotheses can be missed. The set of solutions is generally of any form. Various set-theoretic methods have been developed, each based on a specific representation of measurements and solution: ellipsoid, polytope,

parallelotope or zonotope. We have chosen a representation of variables by intervals. Interval analysis allows an easier implementation and real time computation (Moore 1979).

Sabater and Thomas (1991) is one of the first papers in which set-theoretic approach has been applied to robotic localization for building a geometric representation of the environment. Set-membership localization based on angle measurements has been tackled by Hannebeck and Schmidt (1996), Garulli and Vicino (2001) and based on telemetric data by Lévêque et al. (1997) using SIVIA algorithm described by Jaulin et al. (1993). Di Marco et al. (2004) provide a set-theoretic framework for localization and Simultaneous Localization and Map building (SLAM). Drocourt et al. (2003) apply interval analysis for modeling inaccurate measurements of on-board two omnidirectional sensors for robot localization. Ceccarelli et al. (2006) conclude that simulations and experimental tests show that the set-theoretic approach represents a valuable alternative to statistical SLAM whenever the bounded errors assumption holds on. If the hypothesis is verified, the result is guaranteed. Moreover, as the dimension of the state vector  $(x_R, y_R, \theta_R)$  is equal to three, the data processing is relatively simple and fast.

In fact, little research has investigated the violation of bounded errors assumption for vehicle localization in membership approaches. The respect of the assumption that all the errors are bounded is difficult to demonstrate but there are techniques that permit to reject outliers (Jaulin et al. 2002, Jaulin et al. 2009). Lambert et al. (2009) present a Bounded-Error State Estimation (BESE) for the localization of outdoor vehicle. Authors show that BESE approach is able to solve the localization problem with better consistency than Bayesian approaches such as particle filters, in presence of outliers. Reynet et al. (2009) localized a robot from measurements of ultrasonic sensors treating the outliers under certain conditions. Drevelle et al. (2010) focused on the robustness of set-membership approach in presence of outliers for multi-sensory localization. However, in most works, the rejection of outliers is done by releasing some constraints. This technique can lead to a solution consisting of unconnected sets

In order to propose an alternative to particle filtering, we present a scheme based on bounded-error estimation for robot localization. The approach deals with realistic cases by taking into account model nonlinearities, measurement diversity, process disturbances and measurement errors. In addition the approach handles the presence of outliers and the eventuality of robot kidnapping. Outlier processing is based on constraint relaxation. As mentioned above, the solution could be composed of non-contiguous sets. Therefore we complement the bounded-error estimation scheme by the ability of tracking multiple hypotheses.

The approach offers a well-adapted framework to ambient environment context in which information, useful for the robot localization, can be provided not only by the robot itself but also by the Internet of Things (IoT) present in the environment. Things include for instance the sensors of home automation systems such as infrared detectors or video cameras, or other mobile objects such as another robot or a smartphone.

This paper is organized as follows. Section 2 introduces the theoretical formulation of set-membership approach and then describes the application for localizing a robot from heterogeneous measurements. Section 3 presents complementary methods for handling outliers, asynchronous measurements, environment model inaccuracies and robot kidnapping, in order to meet more realistic conditions. This section ends with the description of the Bounded-Error Estimation Scheme (BEES). Section 4 illustrates the process of localization by BEES step by step from simulated data. Section 5 summarizes recalls the contributions of this work, the limits of the approach and gives information on computing time.

## **2 Set-membership approach for global localization**

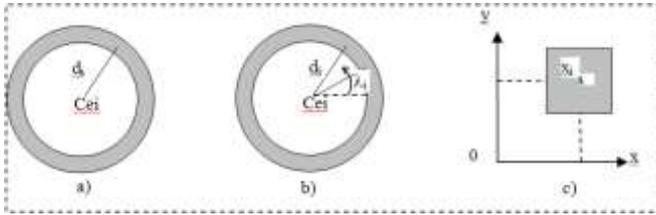
Section 2 introduces the theoretical formulation of the approach and then develops an example of the application for the robot localization from goniometric measurements.

### **2.1 Localization problem**

#### *2.1.1 Global localization problem*

Mobile robot localization consists in the estimation of robot coordinates by using real-time measurements. In ambient environment context, measurements can both come from the robot on-board sensors and from environment objects, mobile or not, able to sense the robot.

The variables to be estimated are the components of the state vector  $\mathbf{x} = (x_R, y_R, \theta_R)^t$  which defines the 2D-position  $(x_R, y_R)$  and the orientation  $\theta_R$  of the robot relatively to the reference frame  $R_e$ . Observations are the set of measurements  $\lambda$  which can be of various types: angle, distance, location or proprioceptive data. Known data are the coordinates of the beacons  $M_j (x_j, y_j)$  and the coordinates and the orientation of environment sensors  $C_j (x_j, y_j, \theta_j)$ . Two cases have to be considered. In one hand, if a robot sensor detects a beacon  $M_i$ , the measurement model depends on the beacon coordinates  $M_i (x_i, y_i)$  and on the state vector. In the other hand, if the robot is detected by an environment sensor  $C_i$ , the measurement depends on the sensor coordinates and orientation  $C_j (x_j, y_j, \theta_j)$  and on the state vector. The known data is supposed to be learned after a learning step not presented in the paper. The state vector  $\mathbf{x} = (x_R, y_R, \theta_R)^t$  is then estimated from the  $n$  observations  $\lambda = (\lambda_1, \dots, \lambda_n)$  associated to  $n$  measurement models and known data  $\mathbf{x}_i = (x_i, y_i)^T$  and  $\mathbf{x}_j = (x_j, y_j, \theta_j)^T$ . Fig. 1 illustrates measurement inaccuracy in a 2D space of sensors frequently used in home automation. Table 1 gives the measurement model of home automation sensors.



**Fig. 1** Measurement inaccuracy. **a** Range. **b** Angle and range. **c** Tactile tile

**Table 1:** Measurement model for various sensors of home automation

Type of measurement	Measurement	Measurement model
Goniometric	angle $\lambda_i$ or $\lambda_j$	$h_i(\mathbf{x}) = \text{tg}^{-1}(\frac{y_R - y_i}{x_R - x_i}) - \theta_i$ or $h_j(\mathbf{x}) = \text{tg}^{-1}(\frac{y_R - y_j}{x_R - x_j}) - \theta_R$
Range (Fig.1a)	range $d_i$	$h_d(\mathbf{x}) = \sqrt{(x_R - x_i)^2 + (y_R - y_i)^2}$
Goniometric and range from the same view point (Fig.1b)	(angle $\lambda_i$ or $\lambda_j$ ) and range $d_i$	$(h_i(\mathbf{x}) \text{ or } h_j(\mathbf{x}))$ and $h_d(\mathbf{x})$
Dead reckoning	$\Delta x_p, \Delta y_p, \Delta \theta_i$	$x_{Rt} = x_{Rt-1} + \Delta x_p, y_{Rt} = y_{Rt-1} + \Delta y_p, \theta_{Rt} = \theta_{Rt-1} + \Delta \theta_i$ at time $t$ and $t-1$ ,
Tactile tile (Fig.1c)	Tile center coordinates : $C_e (x_p, y_i)$	$x_i = x_R, y_i = y_R$
Door crossing detector	$C_{ei}(x_p, y_i)$ (Same case as a narrow tactile tile)	$x_i = x_R, y_i = y_R$
Complex shape	$\{C_{ei}(x_p, y_i)\}$ for $i = 1$ to $k$ . The shape is divided into a set of tactile tiles	It exists $i$ in $1:k / x_i = x_R, y_i = y_R$

Global localization is relevant not only at startup for initializing the state estimate, but also during operation for recovery in case of pose tracking failure.

## 2.1.2 Definitions

### 2.1.2.1 Multihypothesis tracking

Pose tracking consists in maintaining an estimate of the robot pose as the robot moves. An hypothesis  $H_i$  is an estimate of the true pose  $\mathbf{x} = (x_R, y_R, \theta_R)^t$ . Rather than selecting one of them, multihypothesis tracking updates each hypothesis using proprioceptive data. This is called the prediction step. When a new measurement  $\lambda_t$  is available, the correction step is applied to all predicted hypotheses.

Given the true pose of the robot  $\mathbf{x} = (x_R, y_R, \theta_R)^t$   
Each hypothesis  $H_i$  is an estimate of the true pose. The hypotheses are updated by means of proprioceptive data (prediction step). Proprioceptive data are obtained by applying proprioceptive measurements to the evolution model of the robot

$$\mathbf{x}_{t,i} = f(\mathbf{x}_{t-1,i}, \mathbf{u}_{t-1}, \mathbf{v}_{t-1}) \quad (1)$$

where  $\mathbf{x}_{t,i}$  is the predicted hypothesis  $H_i$  at  $t$ ,  $\mathbf{x}_{t-1,i}$  the hypothesis  $H_i$  at  $t-1$ ,  $\mathbf{u}_{t-1}$  the proprioceptive data and  $\mathbf{v}_{t-1}$  the noise associated to the evolution model and proprioceptive measurements.

As soon as a new observation  $\lambda_t$  is available, each predicted hypothesis  $H_i$  is evaluated by using the measurement model  $h(\mathbf{x}_{t,i})$  associated to  $\lambda_t$ .

In the context of interval analysis, the criterion used to validate a predicted hypothesis  $H_i$  is:

$$\lambda_t \cap h(\mathbf{x}_{t,i}) \neq \emptyset \quad (2)$$

When validated, a predicted hypothesis  $H_i$  is corrected this way

$$\mathbf{x}_{t,ic} = \mathbf{x}_{t,i} \cap h^{-1}(\lambda_t) \quad (3)$$

where  $h^{-1}(\lambda_t)$  is the inverse function which gives  $\mathbf{x}_t$  knowing  $\lambda_t$ . In the context of the set-membership approach, eq. 3 is solved by using the inversion set algorithm describes in section 2.2.

The management of the set of hypotheses (creation, weight and deletion) is based on credibility. A credibility  $C_{Ri}$  is associated with each predicted hypothesis  $H_i$ . Predicted hypothesis  $H_i$  sees its credibility varies over time depending on new observations. When a new measurement is available the credibility is managed as follows:

Algorithm	Management of a list of l predicted hypotheses
1	<b>for</b> i = 1 to l;
2	<b>if</b> $\lambda_t \cap h(\mathbf{x}_{t,i}) \neq \emptyset$ , $C_{R,i} = C_{R,i} + 1$ ;
3	<b>else</b> $C_{R,i} = C_{R,i} - 1$ ;
4	<b>if</b> $C_{R,p,i} = 0$ , remove $H_i$ from the list of predicted hypotheses <b>endif</b>
5	<b>endif</b>
6	<b>endfor</b>

A new hypothesis is created by the global localization at initialization or after failure of the pose tracking, and then added to the list of predicted hypotheses. The credibility of a new hypothesis is initialized to the value 1. When there are no new measurement, hypotheses are updated using the robot evolution model. Their credibility does not change.

Credibility is a flexible concept that allows to adopt more or less binding selective strategies. We opted for a strategy that eliminates inconsistent hypotheses by taking into account a memory effect. For example, if the credibility  $C_{Ri}$  of the predicted hypothesis  $H_i$  takes the value 2, hypothesis will have to be inconsistent with measurements during two successive steps of correction so that it is removed from the list.

### 2.1.2.2 Synchronization

The objective of synchronization is to have a sufficient number of measures for the calculation of the state  $\mathbf{x}_t$  of the robot at time  $t$ . The principle consists in completing the set of measurements  $\lambda_t$  by measurements  $\lambda_{t-1}$  acquired at time  $t-1$ . Let the state of the robot  $\mathbf{x}_{t-1}$  at time  $t-1$ , and the relative variation of the state  $\Delta\mathbf{x}_{t-1,t}$  between  $t$  and  $t-1$ , the problem is to incorporate the measurements  $\lambda_{t-1}$  taken at time  $t-1$  with the measurements  $\lambda_t$  taken at time  $t$  into the process of computing the state  $\mathbf{x}_t$  at time  $t$ . In the context of interval analysis, we propose to evaluate

$$(\mathbf{x}_t - \Delta\mathbf{x}_{t-1,t}) \cap \lambda_{t-1} \quad (4)$$

It is important to notice that the synchronization step is applied to the measurements not to the state vectors.

## 2.2 Set Inversion for Estimating Parameters

Interval analysis is based on the idea of enclosing real numbers inside intervals and real vectors inside boxes (Moore 1979). The analysis by intervals consists in representing the real or integer numbers by intervals enclosing them. This idea allows algorithms whose results are guaranteed, for example for solving a set of non-linear equations (Jaulin and Walter 1993).

An interval  $[x]$  is a set of  $IR$  which denotes the set of real interval

$$[x] = \{x \in IR \mid x^- \leq x \leq x^+, x^- \in IR, x^+ \in IR\} \quad (4)$$

$x^-$  and  $x^+$  are respectively the lower and upper bounds of  $[x]$ . The classical real arithmetic operations can be extended to intervals. Elementary functions also can be extended to intervals. Given  $f: IR \rightarrow IR$ , such as  $f \in \{\cos, \sin, \arctan, \text{sqr}, \text{sqrt}, \log, \exp, \dots\}$ , its interval inclusion  $[f]([x])$  is defined on the interval  $[x]$  as follow

$$[x] \rightarrow [f]([x]) = \{f(x) \mid x \in [x]\} \quad (5)$$

A *subpaving* of a box  $[x]$  is the union of non-empty and non-overlapping subboxes of  $[x]$ . A guaranteed approximation of a compact set can be bracketed between an inner subpaving  $X^-$  and an outer subpaving  $X^+$  such as  $X^- \subset X \subset X^+$ . Set inversion is the characterization of

$$X = \{x \in IR^p \mid f(x) \in Y\} = f^{-1}(Y) \quad (6)$$

With  $p$  the dimension of the vector  $x$ ,  $IR^p$  is the set of 1-dimensional boxes and  $Y$  the set to be inverted.

For any  $Y \subset IR^p$  and for any function  $f$  admitting an inclusion function  $[f]$ , two subpavings  $X^-$  and  $X^+$  can be obtained with the algorithm SIVIA (Set Inverter Via Interval Analysis) proposed by Jaulin et al. (1993). To check if a box  $[x]$  is inside or outside  $X$ , the inclusion test  $T$  is composed of two tests:

if  $[f]([x]) \subset Y$  then  $[x]$  is feasible  
 if  $[f]([x]) \cap Y = \emptyset$  then  $[x]$  is unfeasible  
 else  $[x]$  is ambiguous that is feasible or unfeasible

Boxes for which these tests fail are bisected except if they are smaller than the required accuracy  $\varepsilon$ . In this case, boxes remain ambiguous and are added to the  $\Delta X$  subpaving of ambiguous boxes. The outer subpaving  $X^+ = X^- \cup \Delta X$  encloses the solution set  $X$ . The inversion set algorithm can be divided into three steps:

- Select the prior feasible box  $[x_0]$  assumed to enclose the solution set  $X$ ,
- Determine the state of a box, feasible, unfeasible or ambiguous;
- Bisect box for reducing  $\Delta X$ .

Algorithm #1 presents SIVIA based on an inclusion function for building  $X^+$  and  $X^-$ :

Algorithm #1 SIVIA (in : $[x_0], f, Y, \varepsilon$ ; out : $X^+, X^-$ )	
1	<b>if</b> ( $[f]([x_0]) \subset Y$ ), $X^+ = X^+ \cup [x_0]$ ; $X^- = X^- \cup [x_0]$ ; <b>return</b> ; <b>endif</b>
2	<b>if</b> $[f]([x_0]) \cap Y = \emptyset$ , <b>return</b> ; ( $[x_0]$ is unfeasible) <b>endif</b>
3	<b>if</b> ( $\omega([x_0]) < \varepsilon$ ), $X^+ = X^+ \cup [x_0]$ ; <b>return</b> ; ( $[x_0]$ is ambiguous)
	<b>else</b>
4	bisect $[x_0], [x_1], [x_2]$ ; /* bisection divides one dimension of into two equal parts
5	SIVIA (in : $[x_1], f, Y, \varepsilon$ ; out : $X^+, X^-$ ); SIVIA (in : $[x_2], f, Y, \varepsilon$ ; out : $X^+, X^-$ );
	<b>endif</b>

This recursive algorithm ends when the width  $\omega[\mathbf{x}]$  of the box resulting from the bisection step is less than the required precision  $\varepsilon$ . The number  $N$  of bisections is less than

$$N = \left( \frac{\omega(x_0)}{\varepsilon} + 1 \right)^p \quad (7)$$

with  $[\mathbf{x}_0]$  the prior feasible box and  $p$  the dimension of the vector  $[\mathbf{x}]$ . Since in the case of the mobile robot localization the dimension of  $[\mathbf{x}]$  is three, the solution can be computed in real-time (Jaulin et al. 1993).

However the computing time of SIVIA must be reduced in order to be compatible with the management of several hypotheses within the framework of the multihypothesis tracking.

Robot localization can be seen as a constraint satisfaction problem (CSP) which is defined by a set of variables  $X$  linked by a set of constraints  $Co$ . Each variable  $x_i$  belongs to a domain  $D_i$ . In the context of interval analysis the domain  $D_i$  will be defined by the interval  $[x_i]$ . The aim of propagation techniques is to contract the domain of the variables as much as possible without losing any solution (Benhamou 1999). Forward-Backward propagation selects the primitive constraints to be used for optimally contracting the size of the domains. It is based on the Waltz algorithm (Waltz 1972). This contractor reduces the domain of each variable by processing each constraint one by one. This operation is repeated until no more significant contraction can be performed (Casini et al. 2014). This contractor is faster than SIVIA but does not always reach the optimal solution. We use a particular combination of a Forward-Backward contractor and SIVIA. This algorithm is not presented in the paper. The principle is as follows. When no more significant contraction can be performed the solution box is bisected and the same process is repeated on both sub-boxes. In addition the result of FBCB is a 3-dimension box  $([x_R], [y_R], [\theta_R])$  which encloses the outer subpaving  $X^z$  while the result of SIVIA is a subpaving.

### 3 Bounded-error estimation scheme (BEES)

Section 3 presents the bounded-error estimation scheme (BEES) and ends with the description of the modules used either by BEES or to take into account the real constraints imposed by mobile robotics in an ambient environment:

- Unknown initial pose or sudden pose change of the robot;
- Data provided by different sources
  - o Exteroceptive and proprioceptive perception of the robot
  - o Environment objects, mobile or not, able to sense the robot, for instance home automation system or another robot.
- Asynchronous measurements. Indeed, in practice, observations are sampled at different times;
- Some outliers among the measurements;
- Drift and inaccuracy of robot evolution model;
- Environment model inaccuracy. Sensor or beacon coordinates are known in the same frame  $\mathbf{R}(x,y)$  but with a certain inaccuracy.

#### 3.1 Bounded-error estimation scheme (BEES)

Different strategies of localization have been described in literature. Set estimator-predictor filter allows a robust estimation of robot pose on a horizon of data (Kieffer et al. 2000). From the previous vehicle state, the current vehicle state is predicted using the proprioceptive data based on a model of vehicle evolution. Then the predicted solution is contracted with the constraints of measurements acquired between the previous time and the current time. The main problem of this approach is that the solution calculated at a current time depends on all previous state estimations. As a risk is taken on each measurement when selecting the measurement interval the confidence that can be associated to the solution progressively decreases with new estimates. In order to avoid this drawback, Drevelle et al. (2010) propose a state estimation using a limited and slipping horizon of the last vehicle poses. Current vehicle pose is predicted with the history of past proprioceptive data. Then the predicted state is corrected with measures acquired at current time. Possible presence of aberrant data in the vehicle pose history is taken into account. The risk of integrity of the solution is related to the risk associated

with each data. Authors consider that proprioceptive data are guaranteed. Therefore the risk is only related to the previous vehicle poses of history. Using a limited horizon of data provides estimations at constant risk.

Unlike Drevelle et al, we do not assume that proprioceptive data are still valid. To avoid that the current estimation depends on previous pose estimations but only on measurements, we propose to use a sliding history of previous measurements rather than a horizon of previous states. These measures are synchronized at a given time thanks to the history of the proprioceptive data. The goal is to have a sufficient number of measurements to obtain a reliable result while avoiding that the confidence on the result decreases over estimations.

Bounded-error estimation scheme we propose is based on a classical predictor-corrector filter which operates as follows. Pose hypotheses are updated using proprioceptive data. Over time these predicted hypotheses see their credibility varies depending on new observations. This is the classical scheme. In our case the filter considers that the occurrence of disruptive events is possible: the presence of outliers, the sudden change of robot pose, and the drift and the inaccuracy of the robot evolution model. In most works, drift is not taken into consideration. These events lead to a solution which could be false or reduced and even becomes an empty interval. In all these cases the result is no more guaranteed. Outliers concern not only exteroceptive and environment measurements but also proprioceptive data. Exteroceptive and environment outliers make erroneous the correction step of the filter. Proprioceptive outliers or an inappropriate model describing robot evolution make erroneous predicted hypotheses.

### 3.2 Module description

Dead Reckoning is used by BEES for the prediction step. The other modules are used for answering to the real constraints imposed by mobile robotics in an ambient environment:

- Data synchronization;
- Processing of environment model inaccuracies (inaccuracy of sensors and beacon coordinates);
- Outlier processing;
- Multi-hypothesis tracking.

#### 3.2.1 Dead Reckoning for prediction step

Pose tracking provides a new pose estimate, given a previous position and dead reckoning. The model is  $x_{Rt} = x_{Rt-1} + \Delta x_{t-1,t}$ ,  $y_{Rt} = y_{Rt-1} + \Delta y_{t-1,t}$ ,  $\theta_{Rt} = \theta_{Rt-1} + \Delta \theta_{t-1,t}$ . The inclusion test is  $[x_t] \subset [x_{t-1}] + [\Delta x_{t-1,t}]$ .  $[x_t]$  and  $[x_{t-1}]$  are the boxes at times  $t$  and  $t-1$  respectively and  $[\Delta x_{t-1,t}]$  the robot displacement given by dead reckoning between  $t-1$  and  $t$ .

#### 3.2.2 Measurement Synchronization

Proprioceptive information can be used in another way for data synchronization. Consider that at a given time there are not enough measurements for computing a global localization.

Let  $[x_t]$  a box assumed to be the solution at time  $t$ ,  $[\lambda_{t-1}]$  a measurement acquired at time  $t-1$  and  $[\Delta x]_{t-1,t}$  the robot displacement given by dead reckoning between  $t-1$  and  $t$ . Algorithm #2 evaluates if the past state  $[x_{t-1}] = [x_t] - [\Delta x]_{t-1,t}$  is compatible with the past measure  $[\lambda_{t-1}]$ .

Algorithm # 2    Inclusion test ( $[x_t]$ , $[\lambda_{t-1}]$ , $[\Delta x]_{t-1,t}$ )	
1	if ( $[f] ([x_t - \Delta x_{t-1,t}]) \subset [\lambda_{t-1}]$ , $[x_t]$ is feasible ;
2	else if ( $[f] ([x_t - \Delta x_{t-1,t}]) \cap [\lambda_{t-1}] = \emptyset$ ), $[x_t]$ is unfeasible ;
3	else $[x_t]$ is ambiguous ;
4	endif
5	endif

Algorithm #2 does not rely on the previous state estimates but only on the current state, the exteroceptive measurements and the proprioceptive data. In fact, the principle consists in

synchronizing the current state with the past measurement. Current state is evaluated with the measurements acquired at different past times and at time  $t$  so that enough measurements are available for computing robot localization. One other possible use is the tracking of the same beacon by the on-board camera while robot moving. Synchronization allows an increasingly accurate pose only by using the successive measurements provided by the camera.

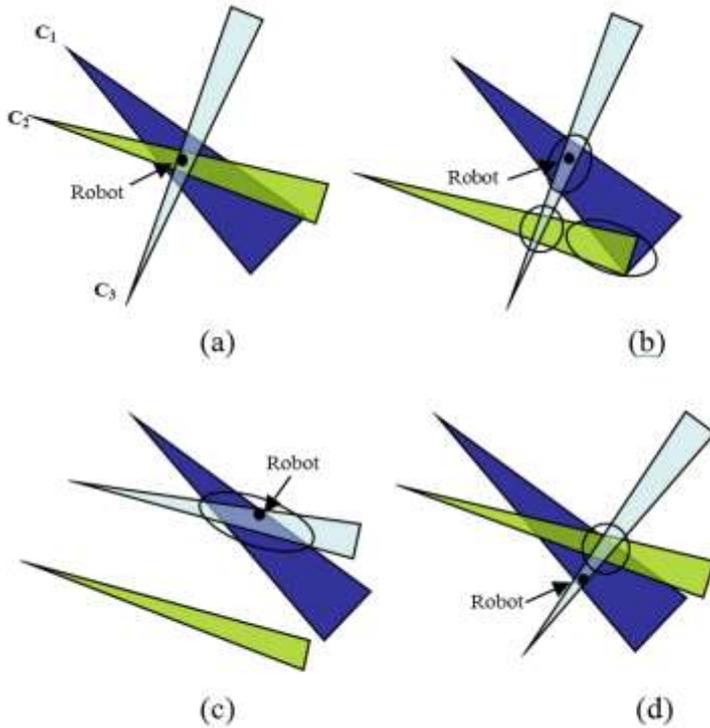
### 3.2.3 Processing of Environment Model Inaccuracies

Up to now we assumed that the coordinates of home sensors and beacons were precisely known. In fact, these devices are approximately located in home frame by the installer of the home automation system. It is easy to consider this inaccuracy by replacing in the measurement equations of sect. 2.2, the scalar value of each coordinate by an interval.  $x_i$  and  $x_j$  respectively become  $[x_i]$  and  $[x_j]$ . FBCB algorithm (see sect. 2.2) is applied to measurement equations in which beacon or sensor coordinates are replaced by intervals. In this case, the algorithm reduces both the robot space vector and the environment model inaccuracies either online, or offline during a learning phase. This is inherent to the process of the forward-backward contractor which is able to decrease all interval domains of each variable, input variable as well as output variable.

### 3.2.4 Outliers processing

Localization is guaranteed assuming bounded-errors. In case of violation of this assumption, the solution could be false, reduced and even become an empty interval depending on the fusion rule of the inclusion tests. This is the case if the width of the interval associated to the measurement is too narrow or if a defective sensor provides an aberrant measurement. Incorrect data which do not respect the bounded-error model are called outliers. Two non-exclusive ways can be followed for processing outliers. The first one consists in better setting bounds of measurements. Bounds might be deduced from the tolerance given by the sensor manufacturer. Nevertheless, the better the process knowledge is the better the determination of bounds. Drevelle et al. (2010) take model errors and risk into account. The choice of bounds associated with each measurement is accompanied by a risk. Given a prior measurement error distribution, the probability that the correct result is inside the solution can be estimated. The method needs a prior measurement error distribution and is only applied to a homogeneous set of GPS measurements. Another way for processing outliers consists in assuming that the set of measurements includes a given number of outliers. Knowing the number  $q$  of outliers, robustness is improved by relaxing  $q$  constraints. The  $q$ -relaxed algorithm computes a solution consistent with at least  $n-q$  measurements where  $n$  is the size of the observation vector.

Detection of aberrant value inside a set of measures is not always trivial. Fig. 2 illustrates the case of robot localization from a set of three goniometric measurements. Measurements are represented by a cone and the robot by a black point. Sensors are labeled  $C_i$ . In Fig. 2a the three measurements are consistent and the solution given by the set algorithm is the intersection of the three cones which includes the true robot localization. In the other three cases, the measurement of sensor  $C_2$  is wrong. In Fig. 2b and Fig. 2c the solution will be empty unless we use the  $q$ -relaxed algorithm assuming one outlier. If  $q = 1$  the solution of Fig. 2b is a set of three subsets which are the intersections of at least two cones (ellipse). Fig. 2d underlines the limit of the  $q$ -relaxed algorithm. The outlier is undetectable because of the consistency of the three measurements. There is a non-empty intersection but the robot localization is wrong unless we assume one outlier.



**Fig. 2** Outlier processing. **a** No inconsistent measurement. **b** One detectable inconsistent measurement, **c** One detectable and identifiable inconsistent measurement. **d** No detectable inconsistent measurement, the true robot localization is not in the intersection area of the three measurements

The main weakness of constraint relaxation is that it generates larger solutions possibly composed of non-contiguous sets. In addition, determining the number  $q$  of supposed outliers presents a difficulty. Jaulin et al. (2002) proposed GOMNE (Guaranteed Outlier Minimal Number Estimator), an algorithm for minimizing the number  $q_{\min}$  of data considered as outliers. The process is too much time consuming for being run on-line but can be used off-line.

### 3.2.5 Multihypothesis tracking

Localization algorithm is a multihypothesis tracking based on predictor-corrector filters, one by hypothesis. Updating process creates, keeps, reduces or kills hypotheses after comparing predictions and measurements at each iteration. In the Internet of Things, the robot goes through sensor-poor areas that alternate with sensor-dense areas. Therefore the number of observations available for localization varies during the robot trajectory. In the localization algorithm hypothesis updating is split into three stages which are activated depending on the number  $n$  of observations available at the current iteration. The correction step of filters is performed at each new observation. If there is no new observation, predicted hypotheses are updated with proprioceptive data. If the number of observations is less than  $(3+q)$  predicted hypotheses are corrected by new measurements. If the number of observations is sufficient a global localization (FBCB) is completed using  $q$ -relaxed intersection (see sect. 3.2.4) similarly, if a disruptive event is detected. However, in this case if the current number of observations is insufficient, global localization is achieved by adding past measurements after synchronizing (see sect. 3.2.2).

## 3.3 BEES Algorithm

Before describing BEES some clarifications have to be recalled:

- A pose hypothesis is a 3-dimension box ( $[x]$ ,  $[y]$ ,  $[\theta]$ ) noted  $[x]$ .
- The result of FBCB algorithm which is called by algorithm BEES, is a list  $L_{Hc}$  of 3-dimension boxes  $[x]_{c,j}$ ,  $j = 1$  to  $N_{Hp}$  which enclose the pose of the robot. This list contains one element if  $q = 0$  and perhaps more if  $q \neq 0$ . The parameters of FBCB ( $[x_0]$ ,  $L$ ,  $q$ ) are the initial box  $[x_0]$  in

- which the robot is supposed to be, the list  $L$  of  $n$  available measurements and the presumed number of outliers  $q$ .
- $L_m$  is the list of  $m$  synchronized measurements. It is a sliding history of the latest measurements and the corresponding proprioceptive data.

*Algorithm description*

Let

- $[\mathbf{x}_0]$ : Initial state space
- $n$ : Number of measurements available at current time  $t$
- $L_c$ : List of  $n$  current measurements
- $m$ : Number of measurements synchronized at current time  $t$
- $L_m$ : List of  $m$  synchronized measurements
- $q$ : Number of supposed outliers
- $[\mathbf{x}]_{p,i}$ : Predicted hypothesis  $i$
- $N_{Hp}$ : Number of predicted hypotheses
- $L_{Hp}$ : List of predicted hypotheses
- $C_{R p,i}$ : Credibility of predicted hypothesis  $[\mathbf{x}]_{p,i}$
- $[\mathbf{x}]_{c,j}$ : Current hypothesis  $j$  computed from available measurements
- $N_{Hc}$ : Number of current hypotheses
- $L_{Hc}$ : List of current hypotheses

As seen in section 3.2.5, BEES algorithm is a multi-hypothesis tracking based on one predictor-corrector filter by hypothesis. Hypothesis updating is split into three stages which are activated depending on the number  $n$  of measurements available at the current iteration.

Algorithm # 3 BEES	
1	<b>if</b> $n = 0$ then update predicted hypotheses with proprioceptive data (see algorithm n° 4)
2	<b>elseif</b> $0 < n < 3+q$ then correct predicted hypotheses by measurements (see algorithm n° 5)
3	<b>else</b> estimate global localization from measurements (see algorithm n° 6)
4	<b>endif</b>

Algorithm # 4 Updating predicted hypotheses	
1	<b>for</b> $i = 1$ to $N_{Hp}$
2	$([\mathbf{x}]_{p,i})_t = ([\mathbf{x}]_{p,i})_{t-1} + [\Delta\mathbf{x}]_{t-1,t}$
3	<b>endif</b>

Where  $([\mathbf{x}]_{p,i})_t$  and  $([\mathbf{x}]_{p,i})_{t-1}$  are respectively the predicted hypotheses  $i$  at  $t$  and  $t-1$ , and  $[\Delta\mathbf{x}]_{t-1,t}$  the robot displacement given by dead reckoning between  $t-1$  and  $t$ .

<b>Algorithm # 5 Correction of predicted hypotheses by measurement(s)</b>	
1	counter = 0;
2	<b>for</b> i = 1 to $N_{Hp}$ , $[x]_c = \text{FBCB}([x]_{p,i}, L_c, q=0)$ ;
3	<b>if</b> $([x]_c \cap [x]_{p,i} = \emptyset)$
4	counter = counter+1;
5	$C_{R_{p,i}} = C_{R_{p,i}} - 1$ ;
6	<b>if</b> $C_{R_{p,i}} = 0$ , remove $[x]_{p,i}$ from the list $L_{Hp}$ ;
7	$N_{Hp} = N_{Hp} - 1$ ;
8	<b>endif</b>
9	<b>else</b>
10	$[x]_{p,i} = [x]_c \cap [x]_{p,i}$ ;
11	$C_{R_{p,i}} = C_{R_{p,i}} + 1$ ;
12	<b>endif</b>
13	<b>endfor</b>
14	<b>if</b> counter = $N_{Hp}$
15	$L_{Hp} = \text{FBCB}([x]_0, L_m, q)$ ;
16	<b>for</b> i = 1 to $N_{Hp}$ , $C_{R_{p,i}} = 1$ ;
17	<b>endfor</b>
18	<b>endif</b>

Line 2: The search space is limited to the predicted hypothesis  $[x]_{p,i}$ . As the number of current measurements is insufficient, the number  $q$  of supposed outliers is zero. Only one hypothesis is returned by FBCB, so  $j=1$  and  $[x]_{c,j}=[x]_c$ .

Lines 14 to 17: if all intersections between predicted hypotheses and current measurements are empty, robot is supposed lost. Then a global localization is needed. The algorithm FBCB searches in all the space  $[x]_0$ . As the number of measurements is insufficient the latest memorized measurements after synchronizing are used by FBCB. A number  $q$  of outliers can be taken into account.

<b>Algorithm # 6 Global localization</b>	
1	$L_{Hc} = \text{FBCB}([x]_0, L_c, q)$ ; // Create a new list of current hypotheses
2	<b>for</b> j = 1 to $N_{Hc}$
3	counter = 0;
4	<b>for</b> i = 1 to $N_{Hp}$
5	<b>if</b> $([x]_{c,j} \cap [x]_{p,i} = \emptyset)$ , counter = counter + 1;
6	<b>else</b>
7	$[x]_{p,i} = [x]_{c,j} \cap [x]_{p,i}$ ;
8	$C_{R_{p,i}} = C_{R_{p,i}} + 1$ ;
9	<b>endif</b>
10	<b>endfor</b>
11	<b>if</b> counter = $N_{Hp}$
12	add $[x]_{p,i}$ to list $L_{Hp}$ ; // Create a new hypothesis
13	$C_{R_{p,i}} = 1$ ;
14	<b>endif</b>
15	<b>endfor</b>

Global localization is performed for initializing or re-initializing the process or when no predicted hypothesis is valid.

## 4 Simulation results

This section analyzes the behavior of the localization algorithm in case of disruptive events: inconsistent measurements, drift and inaccuracy of evolution model, robot kidnapping. The goal is to

present the action of each sub-treatment and then of the complete algorithm BEES. The algorithm is evaluated with simulated data in order to compare its behavior in known and repeatable conditions.

#### 4.1 Evaluation and criteria

Three types of criteria are used for analyzing behaviors of the algorithm: accuracy, confidence and consistency, and then robustness. Localization accuracy might be assessed according to the average or instantaneous error criteria. The average error as the mean square error RMSE (Root Mean Squared Error) reflects the overall performance of the estimator (Li and Zhao 2006). However it is also interesting to measure the accuracy on a temporal horizon to assess the behavior of the estimator in borderline cases. Instantaneous precision criteria are usually Euclidean error, and position and orientation errors.

An estimator must provide a given uncertainty around the estimate. It is the value of this uncertainty that will determine the confidence and the consistency attributed to the estimator. Generally confidence and consistency are evaluated using probabilistic criteria as NCI (No Credibility Index), NEES (Normalized Squared Error Estimation) or NIS (Normalized Innovation Squared) (Bar-Shalom and Li 1993). They may also be studied using the uncertainty bands  $2\sigma$  or  $3\sigma$  or with the uncertainty ellipse describing the same standard deviations  $2\sigma$  and  $3\sigma$  (Lefebvre et al. 2004).

In this work, by analogy, the uncertainty band is defined by the upper and lower bounds of the state estimate minus the corresponding reference state, one band by dimension of the pose hypothesis. Uncertainty bands will be respectively applied to the instantaneous position and orientation errors. The consistency of the estimator is verified if the zero value is always included in the band. It means that the true value is included in the estimation. Confidence is given by the width of the band. The parameter setting of an estimator is the result of a compromise between confidence, consistency and accuracy.

Robustness is studied in borderline cases by means of instantaneous criteria. Robustness can be put into default by the violation of the assumptions such as outliers and model drift.

#### 4.2 Evaluation protocol

##### 4.2.1 Test bed

The global dimensions of the test bed are 10 m x 10 m. The room is equipped with two goniometric sensors which are fixed on the building for perceiving the robot and two markers (beacons). Markers are detected by the goniometric sensor of the robot. The coordinates of the markers are  $M_1(0;5)$  m and  $M_2(10;5)$  m. The coordinates of building sensors are  $C_3(5;0)$  m and  $C_4(0;10)$  m (see Fig. 3).

A measure  $\lambda$  is defined by an interval bounded by the lower and upper limits

$$[\lambda] = [\lambda - \Delta\lambda, \lambda + \Delta\lambda]$$

Building goniometric sensor:  $\Delta\lambda = \pi/144$

Robot goniometric sensor:  $\Delta\lambda = \pi/36$

The variables to be estimated are the components of the state vector  $\mathbf{x} = (x_R, y_R, \theta_R)^t$

FBCB recursive algorithm ends when the length  $\omega[\mathbf{x}]$  of the solution space is less than  $\epsilon$  (see sect. 2.2). Stop condition is ( $\epsilon_x < 0.1$  m) and ( $\epsilon_y < 0.1$  m) and ( $\epsilon_\theta < \pi/36$  rd).

##### 4.2.2 Scenario

Algorithm estimates ten poses  $(x_R, y_R, \theta_R)$  of the robot along the trajectory. The number of poses has been limited for the readability of results. Two disruptive events have been considered, the presence of outliers and the drift of robot evolution model. Results are shown in different forms:

- Figures which illustrate the estimate of robot pose by a box  $(x_R, y_R)$  along the trajectory;
- Figures which show instantaneous position error on  $x_R$ ;
- RMSE and associated standard deviation on  $x_R, y_R$  and  $\theta_R$ .

### 4.3 Evaluation

#### 4.3.1 Evolution model errors

Evolution model is used for updating predicted hypotheses. Few authors take into account the drift of the evolution model and the inaccuracies on proprioceptive data during the prediction step of the estimator. Without correction, drift leads to the fact that the predicted hypothesis no more includes the true pose of the robot.

Two errors have been added to predicted hypotheses for simulating the default of the evolution model.

- Drift of the box center coordinates

$[x]_{d,i} = [x]_{p,i} + 5\%[\Delta x]$  where  $\Delta x$  is the Euclidian distance between two poses.

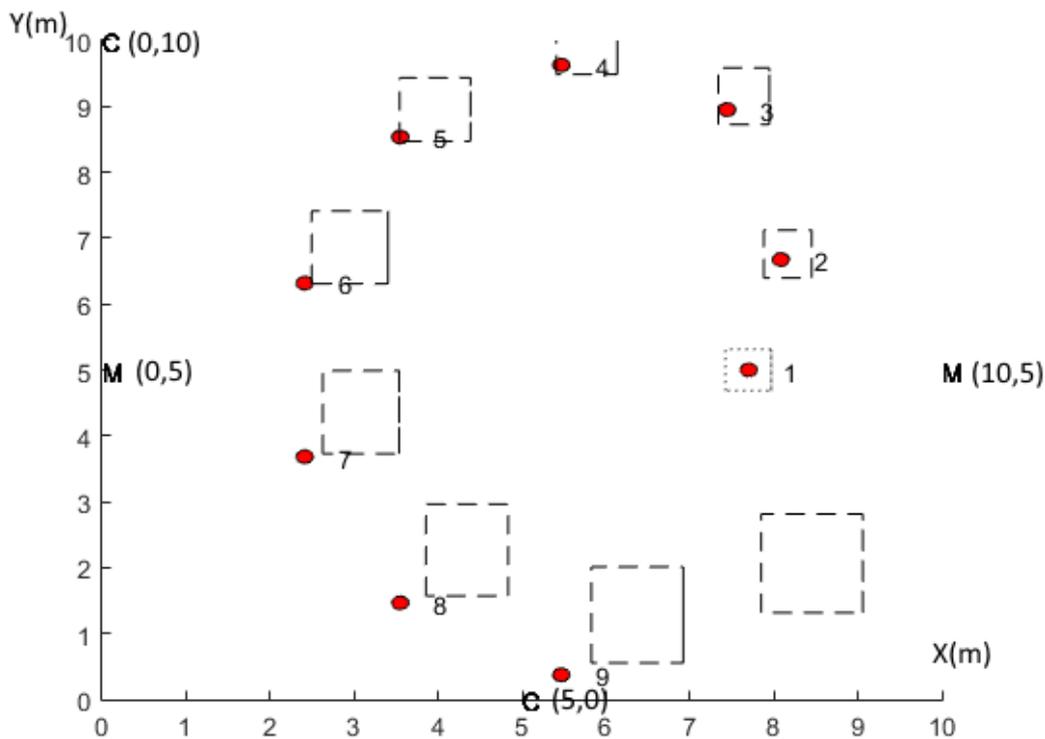
- Cumulative uncertainty by enlarging of hypothesis bounds at each iteration

For lower bounds of  $[x]_{d,i}$

$[x]_{d,i} = [x]_{d,i} - 5\%[\Delta x]$  where  $\Delta x$  is the relative movement of the robot between two poses.

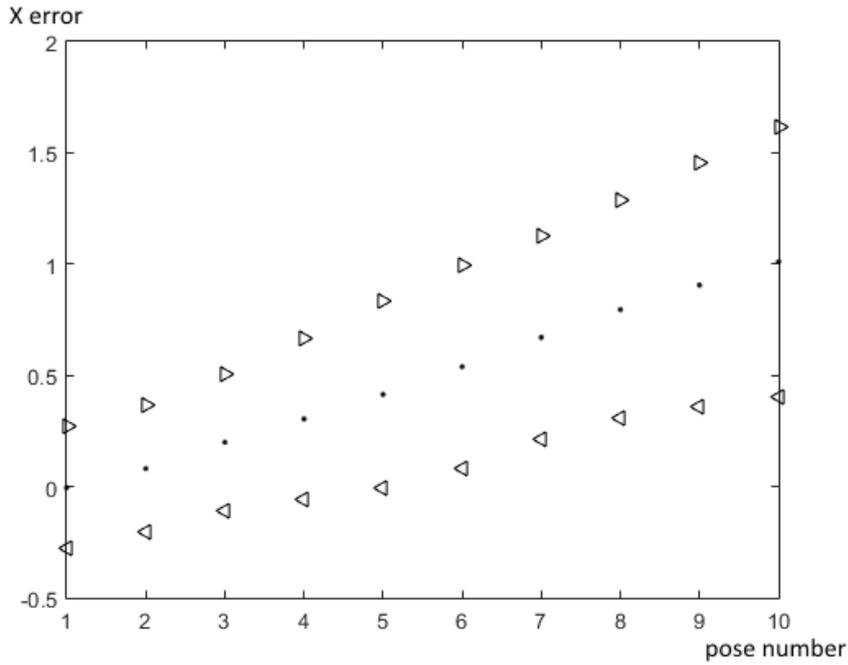
For upper bounds of  $[x]_{d,i}$

$[x]_{d,i} = [x]_{d,i} + 5\%[\Delta x]$



**Fig. 3** Predicted step. Predicted hypotheses (dashed rectangle) of the robot poses along the trajectories in x-y plane (meters). Labels C and M respectively stand for goniometric sensor and marker. Labels are located at the device coordinates. True robot position is represented by a large point

Fig. 3 shows the effect of the defaults of evolution model. At the beginning of the trajectory, the first pose is computed by the global localization algorithm (see # 6 algorithm). From the pose number 6, predicted hypotheses are inconsistent. They no longer include the true position of the robot.

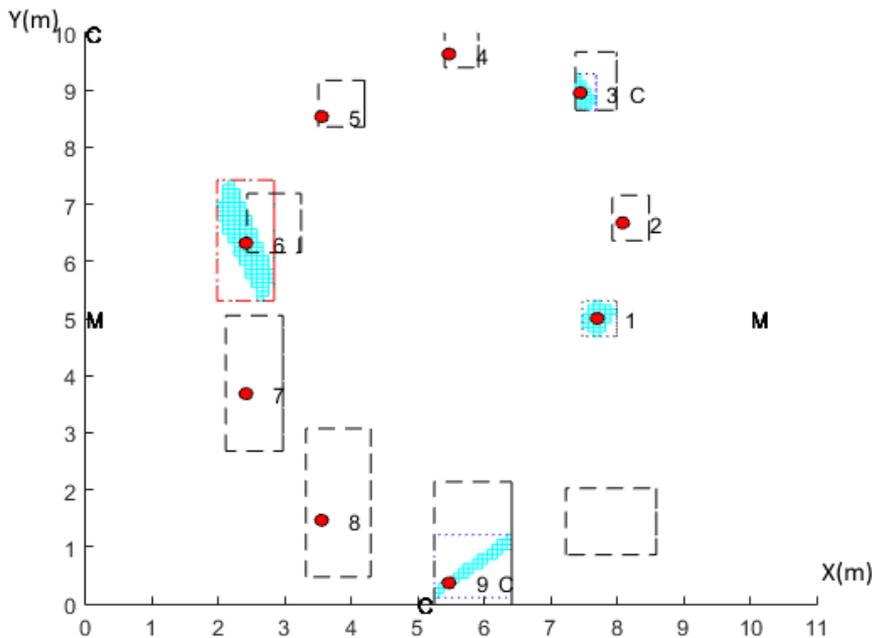


**Fig. 4** Prediction step. Evolution of the instantaneous  $x$  error (black point) in meters with the uncertainty band (black triangle) depending on the number of the pose along the trajectory

Coordinates of the lower bound given at pose number 5 shows that the consistency of the predictor will not be verified from the pose number 6 (see sect. 4.1). Indeed, at pose number 6 the true value  $x$  of robot pose is no more part of the solution (see Fig. 4).

#### 4.3.2 Correction of predicted hypothesis by measurements

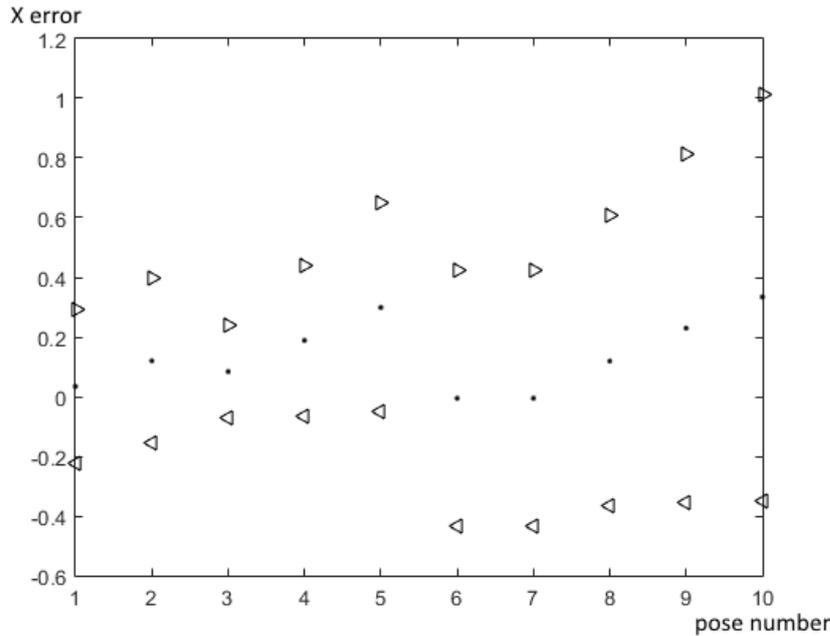
If there are  $n$  available measurements such as  $0 < n < 3+q$  then it is possible to correct predicted hypotheses by measurements (see algorithm n° 5). It is the case of poses number 3 and 9 (see Fig. 5). In the example  $q=0$ .



**Fig. 5** Correction step. Hypotheses of the robot poses along the trajectories in  $x$ - $y$  plane (meters): Predicted hypothesis is represented by dashed rectangle, Corrected hypothesis by dotted rectangle and new hypothesis by dashed-dot rectangle.

Labels C and M respectively stand for goniometric sensor and marker. Labels are located at the device coordinates. True robot position is represented by a large point.

At poses number 1, 3, 6 and 9, the blue subpaving characterizes the common intersection area of the measurements. Subpaving is enclosed by a rectangle. At pose number 6, the intersection between predicted hypothesis and current measurements is empty, not in x-y plane but in  $\theta$  axis. The robot is supposed lost. The global localization FBCB is applied to all the space  $[x_\theta]$  using the latest memorized measurements after a synchronizing step. In the example the size of the stack which memorized past data is four cells. FBCB calculates a new hypothesis represented by the dotted rectangle. This new hypothesis is consistent since it contains the true position of the robot.

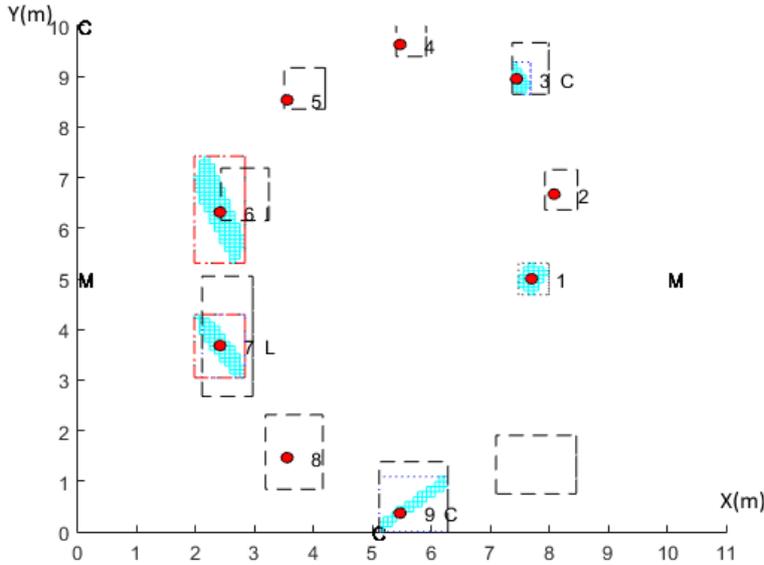


**Fig.°6** Evolution of the instantaneous x error (black point) in meters. The uncertainty band (black triangle) depends on the number of the pose along the trajectory

In Fig. 6 the consistency of the predictor-corrector is now maintained after the pose number 6. For all poses the true value  $x$  of the robot is included in the solution.

#### 4.3.3 BEES

BEES processing depends on the number  $n$  of external measurements available at current time. If  $n \geq 3+q$ , a global localization is computed (poses 1 and 7). If  $0 < n < 3+q$ , predicted hypotheses are corrected by external measurements (poses 3 and 9). If  $n = 0$ , predicted hypotheses are updated using proprioceptive data ((poses 2, 4, 5 and 8).



**Fig. 7** BEES algorithm. Hypotheses of the robot poses along the trajectories in x-y plane (meters): Predicted hypothesis by dashed rectangle, Corrected hypothesis by dotted rectangle and new hypothesis by dashed-dot rectangle. Labels C and M respectively stand for goniometric sensor and marker. Labels are located at the device coordinates. True robot position is represented by a large point

In Fig. 7 new hypotheses (poses 6 and 7), represented by dashed-dot rectangles, correspond to two different cases. At pose number 6, the intersection between predicted hypothesis and measurements is empty (not in x-y plane but in  $\theta$  axis) and the number  $n$  of available measurements is less than  $(3+q)$  (Algorithm #5). At pose number 7, the number  $m$  of available measurements is greater than or equal to  $3+q$  (see Algorithm #6).

Table 2 summarizes the overall performance of each step of the estimator by the RMSE (Root Mean Squared Error) and the standard deviation applied to the state vector.

**Table 2** Average error and the standard deviation of each step of BEES

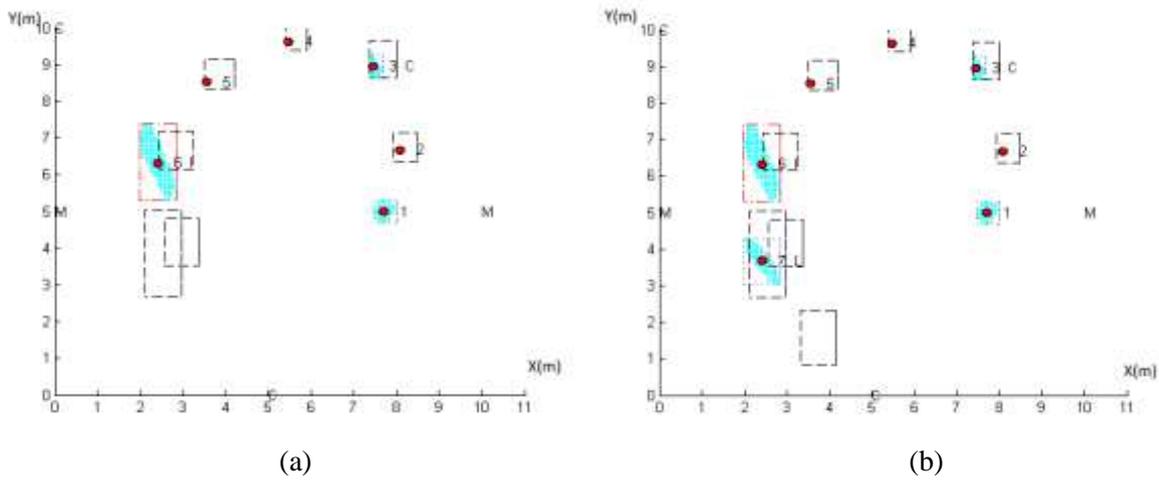
	Algorithm # 4 Prediction		Algorithm # 5 Correction		Algorithm BEES	
	RMSE	Standard deviation	RMSE	Standard deviation	RMSE	Standard deviation
<b>Euclidian distance (m)</b>	0.865	0.489	0.319	0.193	0.229	0.147
<b>x (m)</b>	0.626	0.346	0.240	0.150	0.181	0.121
<b>y (m)</b>	0.626	0.346	0.210	0.135	0.140	0.100
<b><math>\theta</math> (rd)</b>	0.597	0.348	0.180	0.114	0.163	0.119

Regarding computing time, the frequency of BEES algorithm is close to 5 Hz with Matlab software running on an Intel ® Core 2(TM) i7-4710MQ CPU 2.5 GHz computer. Frequency respects the real time constraint of indoor robotics. Results of simulations have been obtained for  $q = 0$ . If we assumed 25% of outliers ( $q=1$ ), the frequency of the algorithm is substantially the same but the localization accuracy decreases: RMSE of  $x = 0.233 \pm 0.187$  m, RMSE of  $y = 0.327 \pm 0.195$  m. The calculation of the RMSE gives a pessimistic evaluation because it is degraded by the drift of the predicted hypotheses and by the incorrect hypotheses when tracking several hypotheses. Therefore, the RMSE of the angle estimate is higher than 0.16 for all algorithms while the sensors error is below 0.087 rad.

#### 4.3.4 Multi-hypothesis tracking

Fig. 8a shows the results of Fig. 7, but incorporating multi-hypothesis tracking. Table 3 shows the evolution of credibility attributed to hypotheses of Fig. 8. At pose number 5 the credibility of predicted

hypothesis (dashed rectangle) is 2 because confirmed twice by external measurements at poses 1 and 3. However this hypothesis is no more confirmed at the pose number 6. The credibility of predicted hypothesis (dashed rectangle) is decremented. As its credibility is 1, the hypothesis remains in the list of predicted hypotheses. At the same pose, algorithm #5 provides a new hypothesis (dashed-dot rectangle) with a credibility initialized to 1. At the new iteration both hypotheses are updated with proprioceptive information (two dashed rectangles).



**Fig. 8** BEES algorithm with multi-hypothesis tracking. **a** At pose number 6. **b** At pose number 7. Hypotheses of the robot poses along the trajectories in x-y plane (meters): Predicted hypothesis by dashed rectangle, Corrected hypothesis by dotted rectangle and new hypothesis by dashed-dot rectangle. Labels C and M respectively stand for goniometric sensor and marker. Labels are located at the device coordinates. True robot position is represented by a large point.

In Fig. 8b the new measurements only confirm the second hypothesis. The credibility of the first hypothesis is decremented and reaches zero. It is removed from the list of predicted hypotheses. The credibility of the second hypothesis is incremented to 2.

**Table 3 :** Evolution of hypotheses credibility for the first eight poses of the trajectory

Pose number	Pose 1	Pose 2	Pose 3	Pose 4	Pose 5	Pose 6	Pose 7	Pose 8
Hypothesis number	1	1	1	1	1	2	2	1
Credibility of the first hypothesis	1	1	2	2	2	1	goes to zero	hypothesis removed
Credibility of the second hypothesis						1	2	2

Table 4 shows the mean of the calculation times of BEES with multihypothesis tracking for the 11 points of the trajectory. Time is calculated from simulated experiments as a function of the precision and the potential number of outliers. For  $q = 1$  we have distinguished two cases: the one for which there is no effective outlier, and the other for which there is one effective outlier.

**Table 4:** BEES computing time with respect to the accuracy and the number of potential outliers

Localization accuracy $\varepsilon$ (m)	0.1	0.05
Computing time (s) with no outlier ( $q=0$ )	0.10	0.34
Computing time (s) with one supposed outlier ( $q=1$ )	0.20	0.61
Computing time (s) with one effective outliers ( $q=1$ )	0.13	0.38

## 5 Experimental results

This section presents the first experiment with a real robot in a real environment. The results give the computing time of FBCB algorithm with respect to the accuracy  $\varepsilon$  and the number  $q$  of potential outliers.

### 5.1 Scenario

The true robot position is  $(3 \text{ m}; 5.7 \text{ m}) \pm 0.2 \text{ m}$  at the center of the triangle defined by three home sensors  $C_j$ ,  $i = 1$  to 3 of goniometric type. Only the robot position  $(x_R, y_R)$  is computed.

### 5.2 Global localization

FBCB computes the robot position using measurements collected by a gateway that handles exchanges with the robot environment. As the robot is detected by a home sensor  $C_j$ , the measurement depends on the sensor position and its orientation  $C_j(x_j, y_j, \theta_j)$ , and the state vector  $(x_R, y_R)$ .

$$\lambda_j = \text{atan2}\left(\frac{y_R - y_j}{x_R - x_j}\right) - \theta_j \quad (8)$$

### 5.3 Test bed

The global dimensions of the test bed are 9.4 m x 6.4 m. Environment is equipped with three goniometric sensors which are video cameras fixed on the top of the walls. Video cameras detect visual markers on the robot. Table 5 gives the testbed equipment specification.

**Table 5:** Testbed equipment specification



Testbed equipment	
Two drive-wheel mobile robot (Droids company)	Environment
2D circular barcode visual markers	Room video cameras, $\Delta\lambda = \pi/72$ ( $2.5^\circ$ )

### 5.4 FBCB computing time

Table 6 gives the computing time with respect to the accuracy  $\varepsilon$  and the number  $q$  of potential outliers with Matlab software (Computer Intel R core 2 duo CPU P9400 2.4 GHz). Only the robot position  $(x_R, y_R)$  is computed.

**Table 6:** FBCB computing time with respect to the accuracy and the number of potential outliers

Localization accuracy $\varepsilon$ (m)	0.5	0.1	0.05	0.01
Computing time (s) with no supposed outlier ( $q=0$ )	0.040	0.049	0.059	0.12
Computing time (s) with one supposed outlier ( $q=1$ )	0.049	0.070	0.109	0.325
Computing time (s) with two supposed outliers ( $q=2$ )	0.079	0.41	0.872	5.096

When assuming no outlier ( $q=0$ ) the measurement frequency is close to 10 Hz for 1cm accuracy. If we suppose two outliers out of three measurements the acceptable accuracy is 0.05 m. It is a promising result easy to improve by using a more efficient programming language. Moreover 66% of inconsistent measurements must be considered as a hard constraint.

## 6. Discussion

### 6.1 Limits of the approach

Some limitations need to be highlighted regarding additional treatments. Synchronization requires the knowledge of robot evolution model. Moreover, as the error due to dead reckoning is cumulative, it is not interesting to synchronize measurements that are too old.

The number  $q$  of supposed outliers is not boundless. Processing time and inaccuracy increase with  $q$ . Experimental simulations have shown that 25 % of outliers seem to be a good compromise between computing time and precision of localization.

## 6.2 Computing time of multitasking algorithm

It depends on three algorithms: Algorithm #5, algorithm #6 and FBCB algorithm. Algorithm #5 runs  $N_{Hp} * FBCB$  where  $N_{Hp}$  is the number of predicted hypothesis. Algorithm #6 runs once algorithm FBCB, and then two nested loops ( $N_{Hc} * N_{Hp}$ ) where  $N_{Hc}$  is the number of current hypothesis and  $N_{Hp}$  the number of predicted hypothesis. Regarding to the number of hypotheses it is important to notice that multihypothesis tracking handles a small number of predicted and current hypotheses. Two arguments explain that fact. First, contrary to particle filter in which a particle corresponds to one pose of the robot, a hypothesis of the set approach can be seen as an infinity of poses uniformly distributed on the interval. It is not necessary to have a great number of hypotheses to represent the potential solution. Secondly, the treatment of outliers creates multiple hypotheses, but the credibility-based hypotheses management quickly reduces the number of hypotheses. In summary BEES computing time mainly depends on FBCB computing time.

## 7. Conclusion

The development of the Internet of Things allows to consider fruitful cooperation between the space of perception of the ambient environment and robots. Set-membership approach based on bounded errors and interval analysis offers a suitable framework for sensor fusion. We have shown that the approach easily takes into account heterogeneous sensors and a variable number of measurements. Measurements are bounded between lower and upper limits. The model is poorer but has the advantage of requiring less statistical knowledge about data. In the paper, the set approach is applied to estimate the robot localization. In interval analysis each hypothesis of robot pose is represented by a 3-dimension box ( $[x_R]$ ,  $[y_R]$ ,  $[\theta_R]$ ). The box dimensions only depend on the measurement context (measurement accuracy, spatial configuration of sensors/beacons and robot). Contrary to most localization algorithms, the set-membership approach is able to give a solution as soon as only one measurement is available. This property is interesting in a poor sensor environment.

The set-membership estimator provides a guaranteed solution if the assumption of bounded error is respected. The violation of this assumption can be avoided by changing the bounds of the interval which encloses the measurement. Another way consists in relaxing  $q$  constraints with  $q$  the number of supposed outliers. However relaxation can lead to a solution composed of non-contiguous sets, each set being considered as a pose hypothesis. Therefore a multi-hypothesis tracking is needed.

The paper addresses the set-membership estimator, the multihypothesis tracking and additional treatments able to respond to more realistic application requirements:

- Measurement inaccuracy;
- Environment model inaccuracy;
- Asynchronous data;
- Disruptive events.

Three disruptive events have been processed. We consider i) outlier as an inconsistent measurement, ii) robot kidnapping as a sudden change of pose which needs a new initialization of the localization process and iii) the drift and inaccuracy of the evolution model of the robot. The prediction step of an estimator depends on this model. Drift of the model between two iterations are often overlooked by the authors. Disruptive events are detected when intersection between predicted and corrective hypotheses is empty during correction step. In any case the behavior of the estimator depends on the number of available measurements  $n$  with respect to the reference value  $(3 + q)$  wherein  $q$  is the number of outliers. Number  $n$  might be increased and observations might be collected at different times thanks to the synchronization mechanism.

From the viewpoint of run time, the frequency of FBCB algorithm, close to 5 Hz, respects the real time constraint of indoor robotics. For instance if the maximum speed of the robot is  $1 \text{ m.s}^{-1}$  the robot displacement between two absolute localizations is only 0.2 m.

In future work we are interested in parameter setting according to the characteristics of the application. In particular the supposed number of outlier  $q$  could be determined on-line by integrating the search of the value of  $q$  inside the process. Indeed potential outliers may be assumed whenever the intersection between the predicted measurements and pose hypotheses is empty. In addition we plan to adjust bounds which enclosed variables, measurements and objet coordinates by a learning step seen as a constraint satisfaction problem.

## References

- Filliat, D., Meyer, J. (2003). Map-based navigation in mobile robots, a review of localization strategies. *Journal of Cognitive Systems Research*, 4(4), 243-282.
- Paull, L., Saeedi, S., Seto, M. & Li H. (2014). AUV Navigation and Localization-ng A review. *IEEE Oceanic Engineering Journal*, 39(1), 131-149.
- Cox, I. J. (1991). Blanche - an experiment in guidance and navigation of an autonomous mobile robot. *IEEE Transactions on Robotics and Automation* 7(3), 193-204.
- Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. F. & Csorba M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation* 17(3), 229-241.
- Leonard, J. J., Durrant-Whyte, H. F. & Cox, I. J. (1992). Dynamic map building for an autonomous mobile robot *IEEE Transactions on Robotics and Automation* 11 (4), 89-96.
- Castellanos, J. M., Neira, J. & Tardos, J. D. (2004). Limits to the consistency of EKF based SLAM. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles* (pp. 1244-1249).
- Piasecki, M. (1995). Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 6, 93-104.
- Thrun S., Fox D., Burgard W. & Dellaert F. (2000). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 99-141.
- Gustafson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 7(25), 53-81.
- Dissanayake, G., Huang, S. & Wang, Z. (2011). A review of recent developments in simultaneous localization and mapping . In *proceeding of IEEE International Conference on Industrial and Information Systems (ICIIS)* (pp 477-482).
- Lambert, E., A., Gruyer, D., Vincke B. & Seignez, E. (2009) Consistent Outdoor Vehicle Localization by Bounded-Error State Estimation. In *Proceeding of International Conference on Robots and Systems (IROS)* (pp. 1211-1216).
- Castellanos, J. M., Neira, J. & Tardos, J. D. (2004). Limits to the consistency of EKF based SLAM. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles* (pp. 1244-1249).
- Piasecki, M. (1995). Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 6, 93-104.
- Seignez, E., Kieffer, E.M., Lambert, A., Walter, E. & Maurin, T. (2009). Realtime bounded-error state estimation for vehicle tracking. *International Journal of Robotics Research*, 28(1), 34-48.
- Moore, R.E. (1979). Method and applications of internal analysis, SIAM, Philadelphia.
- Sabater A., Thomas F. (1991). Set membership approach to the propagation of uncertain geometric information. In *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2718-2723).
- Hanebeck U. D., Schmidt G. (1996). Set theoretical localization of fast mobile robots using an angle measurement technique. In *Proceeding of IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1387-1394).
- Garulli, A. & Vicino, A. (2001). Set membership localization of mobile robots via angle measurements. *IEEE Transactions on Robotics and Automation* 17(4), 450-463.
- Lévêque, O., Jaulin, L., Meizel, D. & Walter, E. (1997). Vehicule localization from inaccurate telemetric data: a set of inversion approach. In *proceeding IFAC Symposium on robot Control*, 1, (pp. 179-186).
- Jaulin, L. & Walter, E. (1993). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4), 1053-1064.
- Drocourt, C., Delahoche, L., Marhic, B. & Brassart, E. (2003). Localization of a robot and guaranteed map building using interval analysis. In *Proceeding of International Conference on Principles and Practice of Constraint Programming* (pp. 17-32).
- Ceccarelli, N., Di Marco M., Garulli, A., Giannitrapani, A. & Vicino, A. (2006). Set membership localization and map building for mobile robots. In *Current Trends in Nonlinear Systems and Control*, pp 289-308. Springer.
- Di Marco M., Garulli A., Giannitrapani A., & Vicino A. (2004). A set theoretic approach to dynamic robot localization and mapping. *Autonomous Robots*, 16, 23-47.
- Casini, M., Garulli, A. & Vicino, A. (2014). A constraint selection technique for recursive set membership identification. In *Proceeding of 19th IFAC World Congress* (pp. 1790-1795).
- Jaulin, L., Kieffer, M., Walter, E., Meizel, D. (2002). Guaranteed Robust Nonlinear Estimation With Application to Robot Localization, *IEEE Transaction SMC, Part C Applications and Review*, 32(4), 254-267.
- Jaulin, L. (2009). Robust set-membership state estimation; application to underwater robotics, *Automatica*, 45(1), 202-206.
- Reynet, O., Jaulin, L. & Chabert, G. (2009). Robust TDOA Passive Location Using Interval Analysis and Contractor. In *Proceeding of International Conference on Radar* (pp. 1-6).
- Drevelle, V. & Bonnifait P. (2010). Robust positioning using relaxed constraint-propagation. In *Proceeding of International Conference on Intelligent Robots and Systems (IROS)* (10, pp. 4843-4848).
- Kieffer, M., Jaulin, L., Walter, E., Meizel, D. (2000). Robust autonomous robot localization using interval analysis *Reliable Computing*, 6(3), 337-362.
- Benhamou, F. Goulard, F., Granvilliers, L. & Puget, J. F. (1999). Revising hull and box consistency. In *Int. Conf. on Logic Programming*, MIT press, 230-244.
- Waltz, D. L. (1972). Generating Semantic Description from Drawings of Scenes with Shadows. *PhD thesis*, Massachusetts Institute of Technology.
- Bar-Shalom Y. & (1998). Estimation and Tracking: Principles, Techniques, and Software. Artech House, Boston, MA, 1993. Reprinted by YBS Publishing.
- Li., X. R. & Zhao Z. (2006) Evaluation of estimation algorithms part i: incomprehensive measures of performance. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4) 1340-1358.
- Lefebvre, T., Bruyninckx H. & De Schutter, J. (2004). Kalman filters for non-linear systems: A comparison of performance. *International Journal of Control*, 77(7) 639-653.